AN OFF-LINE COHERENT FQPSK-B SOFTWARE REFERENCE RECEIVER*

Haiping Tsou, Scott Darden, and Tsun-Yee Yan

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California, USA

ABSTRACT

Using Feher patented Quadrature Phase Shift Keying provides improved spectrum efficiency for high rate telemetry systems. This paper presents an off-line coherent FQPSK-B software reference receiver developed for hardware calibration under contract from the Advanced Range Telemetry (ARTM) program. This receiver is developed in Matlab/Simulink to be consistent with the software capacity available across a number of computing platforms at both ARTM and the Jet Propulsion Laboratory. It also offers a cost-effective approach to demonstrate the advance technologies. The functionality and key features of this receiver (including its internal software transmitter) will be addressed in this paper. Results from end-to-end system simulations are included as examples.

INTRODUCTION

Using Feher patented Quadrature Phase Shift Keying (Feher patented QPSK or FQPSK) improves spectrum efficiency for high rate telemetry systems. An off-line coherent software receiver has been developed for FQPSK-B modulation, a variant in the FQPSK family which includes proprietary designed filtering for additional spectrum containment [1]-[2]. It is intended to be used as (i) a stand-alone receiver offering a platform on which advanced technologies enabling further performance enhancement can be cost-effectively demonstrated, and (ii) a software tool that can be customized for testing and validating various hardware FQPSK-B transceivers under procurement consideration. Because of the testing requirements, the receiver has its own internal reference transmitter and a simple additive white Gaussian noise (AWGN) channel model and, therefore, is capable of performing end-to-end FQPSK-B system simulation and performance evaluation.

The software receiver consists of many functional modules developed in Matlab/Simulink, including differential encoding, FQPSK waveform generation, modulation, channel model, carrier and symbol synchronization, coherent demodulation and detection, differential decoding, and some real-time performance monitors and post-processing test result generators. These modules can be configured differently to carry out specific tasks in each of the receiver's four operating modes. For example, as a

---

* The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology under contract with the National Aeronautic and Space Administration.

stand-alone receiver, it provides off-line demodulation, detection, and decoding of a sampled and digitized FQPSK-B signal from an external source, presumably a hardware transmitter, to attain the detected bit stream. Both demodulation and detection are performed coherently and rely on the receiver's carrier and symbol synchronization loops to generate correct carrier phase reference and symbol timing. The bit error rate (BER) can be measured and compared to that of a hardware receiver when the transmitted bit stream is available for BER calculation. The second mode is for hardware transmitter validation in which the software receiver is able to compare the demodulated hardware-generated signal with a reference signal realized by passing an internally generated baseband signal through the same receiver. The resulting error vector magnitude (EVM) measurement will indicate any hardware problems in the transmitter under test. The third and fourth modes are for internal testing in which the software receiver functions as an end-to-end simulation system conducting BER and/or EVM tests under a set of user-specified conditions, including modulator imbalances and different filter designs. They can be used for trade-off studies on design parameters to realize a specification that meets certain design goals.

The FQPSK-B software receiver is developed on a Sun Workstation by using Matlab version 5.3 with Simulink toolboxes such as Communication Toolbox, DSP Toolbox, and Real Time Workshop. Because of the cross-platform compatibility of Matlab/Simulink, it can run on computers using operating systems such as UNIX, Windows, and MacOS as long as a proper version of Matlab/Simulink is installed. It is also interesting to note that, because of the similarity between FQPSK and other conventional schemes in the QPSK family, the software receiver is flexible enough to handle many other types of QPSK signals or their derivatives with simple modifications.

The functionality and features of this Matlab/Simulink-based software receiver will be described in the next section, followed by a description of each of the four operation modes and their specific configurations for different applications. In addition, results from end-to-end system simulations using the internal reference transmitter are provided as examples, followed by a discussion of future works at the end of this paper.

FUNCTIONAL DESCRIPTION

The software receiver includes two major functional units: the internal reference transmitter and the coherent receiver. The internal transmitter, as shown in Fig. 1, is used to generate reference FQPSK-B

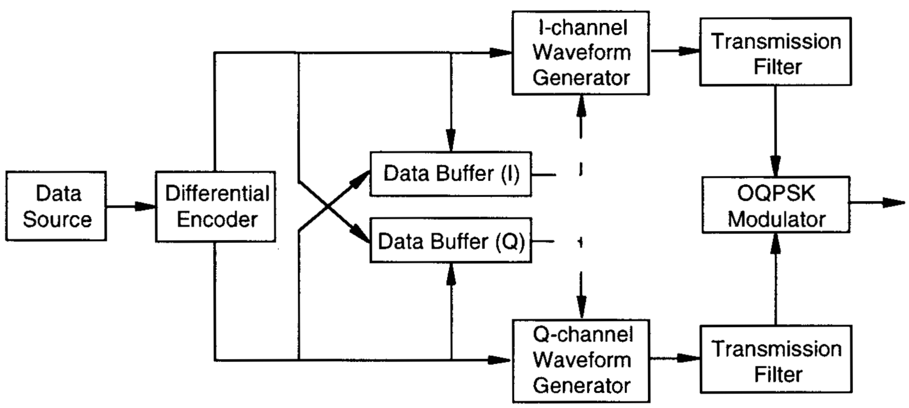


Figure 1 FQPSK-B Software Transmitter

signals for EVM tests and internal end-to-end simulations. It consists of a data source capable of generating either a random or a fixed and known source bit sequence, a differential encoder, an FQPSK waveform generator including the data buffers facilitating cross-correlation between I-channel and Q-channel symbols, two transmission filters for additional spectrum containment, and an offset QPSK (OQPSK) modulator. On the other hand, the coherent receiver, as shown in Fig. 2, consists of an OQPSK carrier tracking loop providing carrier reference for coherent demodulation, a symbol synchronizer providing correct timing for coherent detection, and a differential decoder.

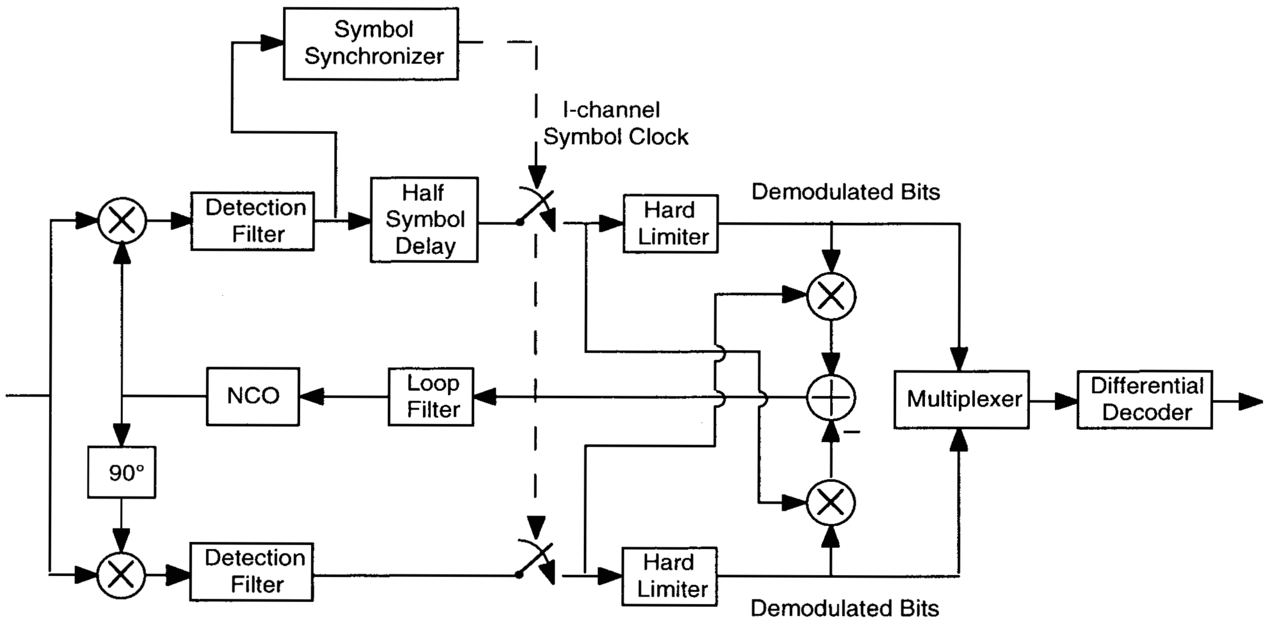


Figure 2 Coherent FQPSK-B Software Receiver

The receiver is developed based upon the assumption that the input signal is an FQPSK-B signal sampled at a fixed (known) rate that is high enough to avoid significant aliasing. The internal transmitter is able to generate such a sampled signal at either baseband or IF with floating-point values. On the other hand, the input from a hardware transmitter under test has to be recorded into a file consists of samples organized according to Matlab's standard I/O format. Typically, these samples are provided by the receiving system's hardware front end[1] that down-converts the received RF signal to IF and then samples it to produce finite-bit quantized samples. These integer-valued samples will be internally converted to floating-point ones by normalizing the average signal power to unity before being processed by the receiver.

Following is the description of the key functions of the transmitter and receiver.

Differential Encoding and Decoding

Differential coding is provided in the software transmitter according to the telemetry standards specified in [3]. Its use is to resolve phase ambiguity associated with QPSK-based modulations.

[1] This is normally a hardware digital oscilloscope capable of down-conversion and sampling.

In the software transmitter, the binary source data stream, consisting of 0's and 1's with a bit duration of T_b seconds, is first differentially encoded. In particular, the k^{th} coded bit, denoted by B_k, is a function of the current source bit and the one before that, denoted by b_k and b_{k-1}, at the encoder input and the $(k-2)^{th}$ coded bit, B_{k-2}, at the encoder output. This function is written recursively as

$$B_k = \overline{b_k \oplus b_{k-1}} \oplus B_{k-2}, \qquad k \in N = \{0,1,2,\cdots\}$$

where the symbol $\oplus$ denotes the exclusive-or operator, and the over-line represents the logical 'not' or inversion operator. The coded bit is then converted from binary format to a non-return-to-zero and leveled (NRZ-L) format by mapping $0 \mapsto -1$ and $1 \mapsto +1$.

The coded NRZ-L bit stream is split into two, with even-numbered bits to be sent through the in-phase (I) channel and odd-numbered bits through the quadrature (Q) channel. On each channel, the coded bit has a duration that is twice the source bit duration because of the splitting of the coded bit stream. It is referred to as the symbol duration, $T_{sym} = 2T_b$, since a QPSK symbol represents two bits, one from each channel.

In the receiver, decoding is performed after demodulation and data detection, rendering the following recursive relationship

$$b_k = \overline{B_k \oplus B_{k-2}} \oplus b_{k-1}, \qquad k \in N$$

between coded and source bits.

Waveform Generation

The generation of baseband FQPSK signal in the internal reference transmitter is based upon a method of performing a symbol-by-symbol cross-correlative mapping on selected I- and Q-channel symbols with a specific set of full-symbol waveforms (wavelets) [4]. Generating an FQPSK-B signal requires additional filtering of the FQPSK signal. In the reference transmitter, a low-pass filter is provided on each of the I- and Q-channels to serve as the transmission filter. The type of filter and its order and cut-off frequency are user-specified parameters.

Modulation and Demodulation

The internal software transmitter can generate both baseband and modulated FQPSK-B signals. According to [3], the Q-channel signal is delayed with respect to the I-channel signal by one source bit, or equivalently a half symbol, resulting in a wavelet-coded baseband OQPSK signal which can be put onto a carrier to produce a modulated FQPSK-B signals at a user-specified frequency. The mapping between a transmitted symbol and the resultant phase is as follows:

I - channel Symbol	Q - channel Symbol	Corresponding Symbol Phase
+1	+1	45°
-1	+1	135°
-1	−1	225°
+1	−1	315°

In addition, the transmitter is capable of emulating an imperfect phase modulator with user-specified modulator imbalances defined in the following equation [5]

$$S(t) = [w_1(t) + a_1]\cos(2\pi f_c t + \theta_c) + G[w_2(t) + a_2]\sin(2\pi f_c t + \theta_c + \Delta\theta)$$

where G is the inter-channel amplitude imbalance, $\Delta\theta$ is the inter-channel phase imbalance, and a_1 and a_2 are I-channel and Q-channel dc biases, respectively. In this modulator imbalance model $w_1(t)$ and $w_2(t)$ represent I-channel and Q-channel signals, f_c is the carrier frequency, and θ_c is a random carrier phase.

On the receiver side, coherent demodulation requires a carrier reference established from the received signal. In particular, the carrier frequency f_c has to be estimated and the carrier phase θ_c has to be continuously tracked. The procedure to get an initial estimate of the carrier frequency and phase is referred to as carrier acquisition. It is done by first creating the carrier components from the received FQPSK-B signal with a full-wave, 4th-law detector [6], and then performing a discrete Fourier transform (DFT) on the detector output to have its frequency domain representation. A simple interpolation on the value of the peak bin around four times the fundamental (carrier) frequency and two values from adjacent bins on both sides of the peak bin produces the estimated frequency and phase.

The tracking of carrier phase is provided by a carrier tracking loop, as shown in Fig. 2, which results from a slight modification of a conventional QPSK cross-over Costas loop [7] by adding a half-symbol delay to its I-arm to compensate for the offset introduced between I- and Q-channel. Immediately after the carrier is acquired, carrier tracking commences with an initialization of the loop's numerically controlled oscillator (NCO) by using the estimated carrier frequency and phase. There are two user-specified low-pass arm filters serving as the detection filters for processing of the FQPSK-B signal. The loop filter implemented is a discrete-time low-pass filter with the following transfer function [8]

$$F(z) = G_1 + \frac{G_2}{1 - z^{-1}} + \frac{G_3}{(1 - z^{-1})^2}$$

where

$$G_1 = rd/T_u$$
$$G_2 = rd^2/T_u$$
$$G_3 = k\,rd^3/T_u$$

and

$$d = \frac{4B_L T_u (r-k)}{r(r-k+1)}$$

with T_u being the loop update interval in seconds, B_L being the loop bandwidth[2] in Hz, $r = 4\xi^2$ where ξ is the damping ratio, and k being the type III loop gain (e.g., $k = 0$ for a type II loop) with typical values ranging from 0.25 to 0.5. By specifying these parameters, a tracking loop with desired loop bandwidth can be exactly realized.

[2] This is the loop bandwidth by design. The actual loop bandwidth may be larger than this designed value depending on the product $B_L T_u$. Generally speaking, for $B_L T_u < 0.05$, the actual loop bandwidth is very close to the designed value.

Coherent Detection

The software receiver performs coherent symbol-by-symbol hard decision separately on demodulated I-channel and Q-channel symbols. As shown in Fig. 2, this is done by sampling the detection filter output at proper times, presumably once at the mid-point of every symbol. Hence, the symbol clock epochs need to be established from the received signal, which is referred to as symbol synchronization. The software receiver uses a data transition tracking loop (DTTL) [9] to provide the estimated symbol timing. It takes the I-arm detection filter output as the input to its two-arm structure, with a full-symbol integration on a symbol and another full- or partial symbol integration centered at the symbol boundary. The latter provides an error signal to drive the estimated clock moving forwards or backwards when the integration window is not exactly centered at a symbol transition. The resulting I-channel symbol clock is at the mid-point of every Q-channel symbol because of the half-symbol inter-channel delay in the FQPSKB signal. It also enables mid-point sampling of I-channel symbols since they get an additional half-symbol delay in the receiver.

SYSTEM OPERATION

The software receiver is composed of four different setups, each being represented by an operating mode in which specific application of the software receiver is addressed with its own configuration. This section provides a description of these four operation modes, namely the receiver mode, the EVM mode, and internal end-to-end BER and EVM modes.

Receiver Mode

This is the mode in which the software receiver functions as an offline coherent receiver that demodulates, detects, and decodes the received signal. In this mode the receiver accepts input from external source without using either of its internal software transmitters. Depending upon whether the recorded input is on a carrier, the operator can turn on or off the carrier acquisition and tracking subsystems. Other receiver parameters the operator can specify include

- data (bit) rate,
- window, DFT size, and zero-padding (if used) for DFT-based carrier acquisition,
- type, order, and BT_b-product of the detection filters,
- order, damping ratio, update interval, and loop bandwidth for carrier tracking loop,
- order, damping ratio, update interval, and loop bandwidth for symbol tracking loop.

During the real-time processing, the operator can monitor the progress by looking at a sample counter showing the number of samples being processed so far, and the tracking performance by bringing up the plots of residual phase and tracking variance versus time for carrier and/or symbol tracking. Besides these real-time indicators, the receiver will store the following data:

- detected soft I-channel and Q-channel samples (I/Q not aligned),
- detected soft I-channel and Q-channel samples (I/Q aligned),
- decoded bits,

into separate files for post processing. The detected I/Q-channel samples are the samples obtained by sampling the demodulated I/Q signal at the estimated symbol clock epochs provided by the symbol tracking. They are 'soft' since they are not passed through the hard decision yet. These sample files can be used to plot phasor diagrams for demodulated symbols before or after the one-bit I/Q-channel offset is removed. The decoded bit stream is the primary product of this mode. BER calculation can be performed on it if the transmitted bit stream is available for comparison, which requires a correlative matching process between the decoded and the source bit streams in the post processing to make them aligned.

EVM Mode

This is the mode for hardware transmitter validation in which the signal from external hardware transmitter is demodulated, detected, and compared to the reference signal from the internal *continuous-time domain* software transmitter. The resulting error vectors provide indication of hardware problems in the transmitter under test. The setup for EVM is divided into two sides. The reference side is a baseband end-to-end simulation system using the continuous-time domain transmitter to generate baseband FQPSK-B signal from a given periodic source bit stream used also on the test unit side. This baseband signal is then filtered by the detection filter of the software receiver, producing output samples to be saved into a reference sample file for later comparison[3]. On the test unit side, the same software receiver used on the reference side coherently demodulates the recorded IF signal from the hardware transmitter under test, rendering a test sample file at its detection filter output. This test samples are aligned with the reference samples by performing a correlative matching process between them. The EVM takes place at ideal symbol clock epochs that are readily available from the reference side by simply counting samples in the reference sample file.

The operator is able to specify receiver parameters other than those of detection filter, which remain fixed for both sides of EVM setup, and get real-time performance monitoring as in the receiver mode. Two data files containing entire I-channel and Q-channel samples after the one-bit I/Q-channel offset is removed will be saved for post processing.

The EVM is done in the post processing, in which a correlative matching is performed to align the I/Q-channel sample files with the reference sample files using either I-channel or Q-channel samples and the aligned files are sampled at ideal detection times established by a sample count on the reference side to get the detected symbols. A resulting EVM file will be created, from which EVM statistics, including mean and standard deviation on magnitude and phase for all error vectors and for error vectors associated with individual symbols, are calculated and the following plots:

- Constellation plot -- a plot of error vectors on the signal constellation
- Normalized plot -- a scatter plot of error vectors in the vector space
- I/Q-channel magnitude errors vs. time

can be produced. In addition to EVM statistics and plots, phasor diagram (only with I/Q samples aligned) is also available in the post processing,

Internal BER Mode

This is the mode for internal BER test in which the software transmitter-receiver pair functions as an end-to-end simulation system conducting BER test under a set of user-specified conditions. It differs

[3] Getting a valid comparison requires the reference sample file to be generated by the internal software transmitter with parameters matching those used in the hardware transmitter under test. In addition, the detection filter has to be specified for this and will also be used on the reference side.

from the receiver mode in the choice of signal source. In this mode, the internal discrete-time-domain software transmitter is used to provide FQPSK-B signal either on a carrier or on baseband. A channel model is included in this end-to-end system. Currently, only an AWGN channel with a user-specified bit signal-to-noise ratio (SNR), denoted by E_b/N_o, is provided. More sophisticated channel models, including frequency selective multiple-way fading channel, can be easily added later.

The operator can specify transmitter parameters such as

- data (bit) rate -- may include an offset,
- carrier frequency -- may include an offset,
- carrier phase,
- type, order, and BT_b-product of the transmission filters,
- source bit stream from a file or a noise seed if internal binary random number generator is used to generate source bit stream,

the channel parameter, E_b/N_o, and the receiver parameters such as

- data (bit) rate,
- window, DFT size, and zero-padding (if used) for DFT-based carrier acquisition,
- type, order, and BT_b-product of the detection filters,
- order, damping ratio, update interval, and loop bandwidth for carrier tracking loop,
- order, damping ratio, update interval, and loop bandwidth for symbol tracking loop.

Furthermore, the discrete-time-domain transmitter is capable of accommodating modulator imbalances such as inter-channel amplitude and phase imbalances and I- and Q-channel dc biases, which allows the effect of modulator imbalances to be included with an end-to-end simulation.

The operation in this mode is all real-time. As in previous two modes, an operator can get real-time performance monitoring, including a BER counter showing coded and uncoded error performance.

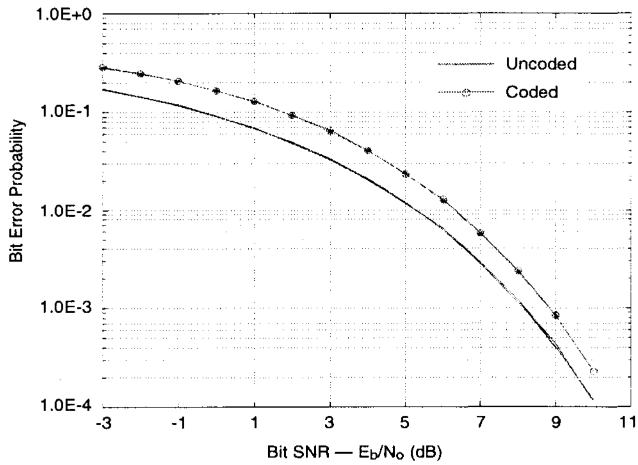
Internal EVM Mode

This is the mode for internal EVM test in which two software transmitter-receiver pairs form an end-to-end simulation system conducting EVM test under a set of user-specified conditions. Like the EVM test for external hardware transmitters, the two software transmitter-receiver pairs in the internal EVM test setup constitute two sides: one for the reference and the other for the test unit. However, in this internal mode, the discrete-time-domain software transmitter provides both the baseband signal for the reference side and the modulated signal for the test unit side. Both sides are synchronized so that EVM calculation can be performed in real-time without the need of a post-processing correlation to align signals from both sides. This setup is intended for simulations to see the effect of (i) user-specified modulator imbalances added to the test unit, (ii) mismatched transmission filters, and (iii) receiver's demodulation and synchronization processing on the EVM results. An operator is able to monitor real-time performance, including selected EVM plots, with EVM statistics reported at the end of simulation.

EXAMPLES

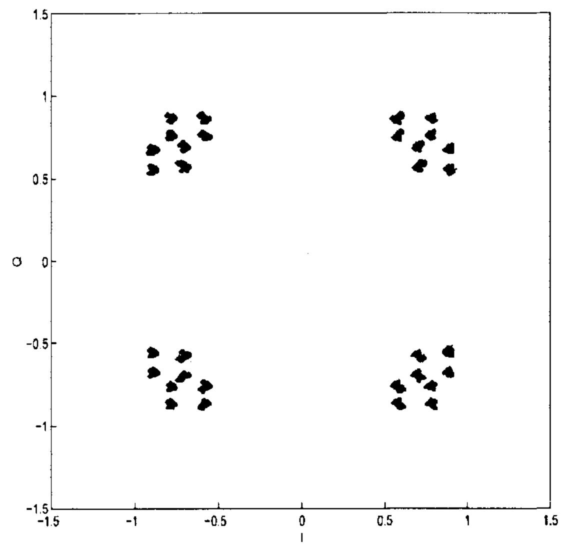
The Bit Error Performance

This example shows the bit error performance for a baseband end-to-end system operated in the internal BER mode. Both coded and uncoded BER curves are shown here for bit SNR, E_b/N_o, from –3 to 10 dB. The uncoded BER curve agrees well with the one reported in [10]. As shown here, the degradation caused by differential encoding/decoding is about 0.63 dB at BER=0.001. Simulation results obtained for systems including carrier and symbol tracking show no significant differences compared to these baseband curves, as long as sufficient loop SNRs are maintained.

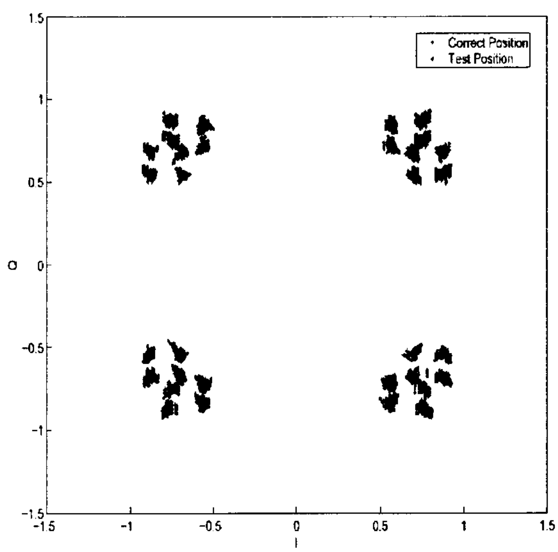


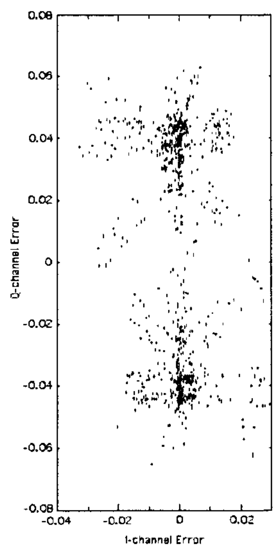
The EVM Test

This example shows the results of an internal EVM test. The phasor diagram to the right is a constellation plot of the perfectly detected (reference) symbols from the receiver after the one-bit I/Q-channel offset is removed. In each quadrant, the corresponding FQPSK-B symbols form a distinct eight-cluster pattern. In the test unit, an amplitude imbalance is deliberately introduced to the Q-channel, making the amplitude of Q-channel symbols 5% smaller than that of I-channel symbols. After demodulation and detection, the error vectors are calculated by comparing both symbol streams symbol by symbol. In the following, two EVM plots are presented as an example. The constellation plot is the figure to the left in which each error vector is represented by an arrow pointing out from its



corresponding reference symbol in the phasor diagram, while the normalized plot is the figure to the right that shows a scatter plot of end point of each error vector in a vector space. It is clearly indicated in these EVM plots that the amplitude imbalance exists in the transmitter under test by showing most arrows pointing roughly along the Q-channel direction in the constellation plot and a two-cluster distribution with an average separation approximately equal to 0.04 in the normalized plot.





CONCLUSION

This paper presents the design and development of an off-line coherent FQPSK-B software reference receiver capable of being used for hardware validation and for advance technology development and demonstration. This Matlab/Simulink-based software receiver also includes end-to-end simulations that help a system design to meet certain performance goals. Because of its ease of use and versatility in flexible configurations for various tasks and even for other types of QPSK modulation, this software receiver provides a cost-effective approach with cross-platform compatibility for study, design and development of a quadrature-modulated communication system.

Currently, the software receiver is being modified to accommodate future studies, including a frequency selective dynamic fading channel model and various equalizer techniques for mitigating the mobile fading effects.

REFERENCES

[1] K. Feher et al.: US Patents 4,567,602; 4,644,565; 5,491,457; and 5,784,402, post-patent improvements and other U.S. and international patents pending.

[2] "FQPSK-B, Revision A1," Digcom-Feher Patented Technology Transfer Document, Digcom Inc., January 15, 1999.

[3] "IRIG Standard 106-00: Telemetry Standards," Telemetry Group, Range Commanders Council, U.S. Army White Sands Missile Range, New Mexico, January 2000.

[4] M. K. Simon and T.-Y. Yan, "Cross-correlated Trellis Coded Quadrature Modulation," U.S. patents pending.

[5] H. Tsou, "The Combined Effect of Modulator Imbalances and Amplifier Nonlinearity on the Performance of Offset Quadrature-Phase-Shift-Keyed (OQPSK) Systems," *The TMO Progress Report 42-137*, Jet Propulsion Laboratory, Pasadena, California, May 15, 1999.

[6] W. B. Davenport, Jr. and W. L. Root, *An Introduction to the theory of Random Signals and Noise*, McGraw-Hill, New York, 1958.

[7] M. K. Simon, "On the Optimality of the MAP estimation Loop for Carrier Phase Tracking BPSK and QPSK Signals," *IEEE Trans. on Communications,* vol. COM-27, No. 1, January 1979.

[8] A. Blanchard, *Phase-Locked Loops: Application to Coherent Receiver Design*, John Wiley & Sons, Inc., New York, 1976.

[9] W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[10] T.-Y. Yan, *Advanced Range Telemetry Task Interim Report*, Jet Propulsion Laboratory, Pasadena, California, February 1999.